

# PriceFinder

**Tu ci dici cosa cercare.  
Noi te lo troviamo.  
Al miglior prezzo!**

---

Docente  
Gennaro Costagliola

Studenti  
Antonio Esposito  
Giuseppe Luciano  
Marco Procaccini

# Indice

- Descrizione del problema: perché usare PriceFinder
  - Architettura
  - Fonti
  - Datalog Fonti
  - Descrizione wrapper
  - Schema globale
  - Query Globali
  - Lav & Gav
  - Tecnologie utilizzate
-

# Descrizione del problema

## Perché usare PriceFinder

- Interfaccia user-friendly per la ricerca di uno smartphone al miglior prezzo



# Descrizione del problema

## Perché usare PriceFinder

- Interroga 5 fonti diverse per sfruttare la concorrenzialità del mercato

The logo for WireShop, featuring the word "WIRE" in blue and "SHOP" in white letters inside blue circles.The logo for MediaWorld, with "Media" in red, a globe icon, and "World" in red. Below it, the word "FOREVER" is written in small black letters.The logo for MobileShop, featuring a speech bubble icon with the letter "M" inside, followed by the word "MOBILESHOP" in black.The logo for PLXmania.com, with "PLX" in green and "mania.com" in black.The logo for Stockisti, featuring a purple silhouette of a muscular man with a white "S" on his chest, next to the word "Stockisti" in white. Below it, the text "abbiamo i #superprezzi!" is written in small white letters.

# Descrizione del problema

## Perché usare PriceFinder

- E' presente lista dei prodotti più cliccati dagli utenti
- I risultati vengono mostrati con:
  - I link raggruppati in base alle caratteristiche del cellulare
  - Le video-recensioni da YouTube
  - Una carrellata di prodotti della stessa fascia di prezzo



IPHONE 6S 128GB

Apple iPhone 6s: la recensione di HDBlog.it

908€ (MOBILESHOP: Apple iPhone 6s 128GB Oro)  
908€ (MOBILESHOP: Apple iPhone 6s 128GB Argento)  
999€ (MEDIOWORLD: APPLE iPhone 6S 128GB Oro Rosa)

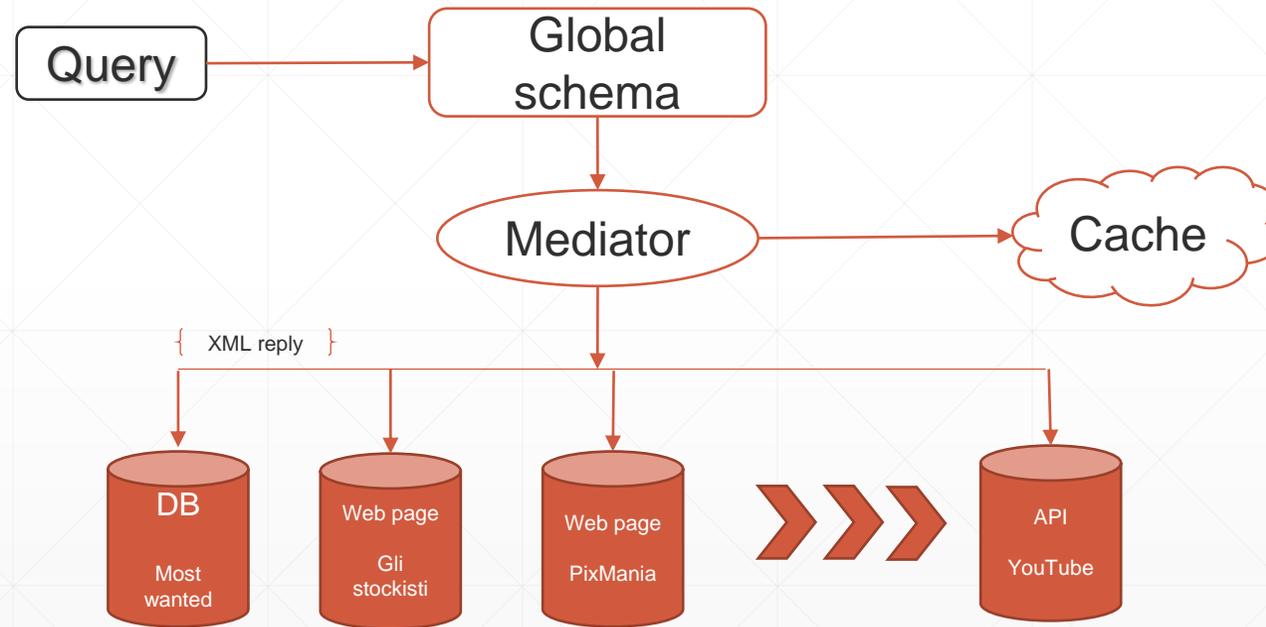
## Più cliccati

- SAMSUNG SM-N910F Galaxy Note 4 White - 499,99€
- Apple iPhone 6 Plus 128GB Oro - 793€
- Nokia Lumia 510 Giallo - 129€
- Huawei Honor 6 Plus 32GB Dual SIM PE-TL10 Bianco - 383€
- LG G4 H815 Bianco Oro - 445€

## Prodotti simili

- 
-  Samsung Galaxy S6 S VI SM-G920F Gold Platinum 32GB a 425,00€
  -  HTC One A9 16GB Opal Silver a 439,00€
  -  Huawei P8 Max 64GB Dual Titanium Grey a 489,00€

# Architettura



# Fonti

- Le fonti utilizzate per ora sono:
    1. 5 pagine web su cui viene fatto scraping delle info desiderate
      - <http://www.stockisti.com/it/>
      - <http://www.mediaworld.it/>
      - <http://www.pixmania.it/index.html>
      - <http://www.wireshop.it/>
      - <http://it.mobileshop.eu/>
    2. Un database relazionale per registrare i click degli utenti
    3. Le API di YouTube per la ricerca delle video-recensioni
-

# Datalog Fonti

- `wireshop(nome,link,immagine,prezzo,fonte)`
  - `stockisti(nome,link,immagine,prezzo,fonte)`
  - `mediaworld(nome,link,immagine,prezzo,fonte)`
  - `pixmania(nome,link,immagine,prezzo,fonte)`
  - `mobileshop(nome,link,immagine,prezzo,fonte)`
  - `db(nome,timestamp,numClick,fonte)`
  - `youtube(nome,linkVideo,regionCode,maxResult)`
-

# Descrizione delle fonti

## Le pagine web

- Per ogni fonte è stato creato uno script php che nell'ordine:
  1. Contatta la pagina remota, cercando il prodotto desiderato
  2. La salva in un oggetto di tipo DomDocument
  3. Estrae, tramite xpath le informazioni dei primi 5 risultati ( se presenti )
  4. Prepara i dati per la risposta in formato XML
  5. Fornisce il risultato della query iniziale

Fonti diverse avranno path delle query diverse, ma la struttura sarà la stessa.

Vediamo ora queste fasi più nel dettaglio, attraverso i punti comuni a tutte.

---

# Descrizione delle fonti - le pagine web

## 1. Contattare la pagina remota

Viene fatto uso della libreria CURL

Standard da PHP  $\geq$  4.0.3

Perfetta per richiedere una pagina web

e gestirne le risposte

```
38 function get_web_page( $url )
39 {
40     $options = array(
41         CURLOPT_RETURNTRANSFER => true,           // return web page
42         CURLOPT_HEADER         => false,         // don't return headers
43         CURLOPT_FOLLOWLOCATION  => true,         // follow redirects
44         CURLOPT_ENCODING       => "",           // handle all encodings
45         //CURLOPT_USERAGENT     => "spider", // who am i
46         CURLOPT_AUTOREFERER    => true,         // set referer on redirect
47         CURLOPT_CONNECTTIMEOUT => 120,         // timeout on connect
48         CURLOPT_TIMEOUT        => 120,         // timeout on response
49         CURLOPT_MAXREDIRS      => 10,         // stop after 10 redirects
50         CURLOPT_SSL_VERIFYPEER => false        // Disabled SSL Cert checks
51     );
52
53     $ch      = curl_init( $url );
54     curl_setopt_array( $ch, $options );
55     $content = curl_exec( $ch );
56     $err     = curl_errno( $ch );
57     $errmsg  = curl_error( $ch );
58     $header  = curl_getinfo( $ch );
59     curl_close( $ch );
60
61
62     $header['errno']   = $err;
63     $header['errmsg']  = $errmsg;
64     $header['content'] = $content;
65     return $header;
66 }
```

# Descrizione delle fonti - le pagine web

## 1. Contattare la pagina remota - 2. Salvarla in un oggetto di tipo DomDocument

- A seconda della pagina da interrogare viene composta la query, in questo esempio la fonte era Gli Stockisti
- Viene quindi caricata la pagina in un oggetto PHP di tipo DomDocument

```
83 // faccio la query e metto la pagina html nella variabile $pagina
84 $pagina = get_web_page("http://www.stockisti.com/it/catalogsearch/result/index/?cat=14&dir=
85 desc&disponibile=1067&index=mage_catalog_product&limit=36&order=price&q=".$product);
86
87
88 // da qui inizia la sezione relativa alla query
89 $dom = new DomDocument();
90
91 @$dom->loadHtml($pagina['content']);
92 $xpath = new DomXpath($dom);
93
```

# Descrizioni delle fonti - le pagine web

## 3. Estrazione delle informazioni

1. Viene creato un array che conterrà le informazioni su:

- Nome
- Prezzo
- Immagine del prodotto
- Link all'offerta

2. Vengono contati i prodotti trovati

3. Per ognuno di essi vengono inserite le relative info nell'array

*continua ...*

---

# Descrizione delle fonti - le pagine web

## 3. Estrazione delle informazioni: fonte Gli Stockisti

```
100 $nomi = array_fill(0, 5, null);
101 $prezzi = array_fill(0,5,null);
102 $immagini = array_fill(0,5,null);
103 $querys = array_fill(0,5,null);
104 $c = 1;
105 $count = $xpath->query("//*[@id=\"root-wrapper\"]/div/div/div[2]/div[2]/div/div[3]/div[3]/ul/li");
106 $count = $count->length;
107
108
109
110
111 for($i =1;$i<= $count;$i++)
112 {
113
114     $nomi[$i] = $xpath->query("//*[@id=\"root-wrapper\"]/div/div/div[2]/div[2]/div/div[3]/div[3]/ul/li[\".$i.\"]/div[2]/h2/a/@title");
115     $prezzi[$i] = $xpath->query("//*[@id=\"root-wrapper\"]/div/div/div[2]/div[2]/div/div[3]/div[3]/ul/li[\".$i.\"]/div[2]/div[3]/span/span/text()");
116     $immagini[$i] = $xpath->query("//*[@id=\"root-wrapper\"]/div/div/div[2]/div[2]/div/div[3]/div[3]/ul/li[\".$i.\"]/div[2]/div[2]/a/img/@src");
117     $querys[$i] = $xpath->query("//*[@id=\"root-wrapper\"]/div/div/div[2]/div[2]/div/div[3]/div[3]/ul/li[\".$i.\"]/div[2]/h2/a/@href");
118     $nomi[$i] = $nomi[$i]->item(0)->nodeValue;
119     $prezzi[$i] = $prezzi[$i]->item(0)->nodeValue;
120     $immagini[$i] = trim($immagini[$i]->item(0)->nodeValue);
121     $querys[$i] = trim($querys[$i]->item(0)->nodeValue);
122
123 }
```

# Descrizione delle fonti - le pagine web

## 3. Estrazione delle informazioni: fonte Pixmania

```
$nomi[$i] = $xpath->query("//*[@id=\"content\"]/div[2]/section/form/div[2]/div/article[\".$i.  
\"]/div[2]/div[1]/header/a/@title");  
$prezzi[$i] = $xpath->query("//*[@id=\"content\"]/div[2]/section/form/div[2]/div/article[\".$i.  
\"]/div[2]/div[2]/a/span[@class=\"currentPrice\"]/ins");  
$immagini[$i] = $xpath->query("//*[@id=\"content\"]/div[2]/section/form/div[2]/div/article[\".$i.\"]/a/img[2]/@src");  
$queries[$i] = $xpath->query("//*[@id=\"content\"]/div[2]/section/form/div[2]/div/article[\".$i.  
\"]/div[2]/div[1]/header/a/@href");
```

---

# Descrizione delle fonti - le pagine web

## 3. Estrazione delle informazioni: fonte Wireshop

```
$nomi[$i] = $xpath->query("//table//h2/a [\".$i.\"]");  
$immagini[$i] = $xpath->query("//table//td[@class='picture'] [\".$i.\"]/a/img/@src");  
$prezzi[$i] = $xpath->query("//table//td/span[@class='price'] [\".$i.\"]");  
$querys[$i] = $xpath->query("//table//h2/a [\".$i.\"]/@href");
```

---



# Descrizione delle fonti - le pagine web

## 3. Estrazione delle informazioni: fonte MobileShop

```
$nomi[$i] = $xpath->query("(//html/body/div/div/div/section/div/ul/li[div[@class='price' and text() != 'Esaurito' and text() != 'Fuori Produzione']])[".$i."]/a");  
$prezzi[$i] = $xpath->query("(//html/body/div/div/div/section/div/ul/li[div[@class='price' and text() != 'Esaurito' and text() != 'Fuori Produzione']])[".$i."]/div[@class='price']");  
$immagini[$i] = $xpath->query("(//html/body/div/div/div/section/div/ul/li[div[@class='price' and text() != 'Esaurito' and text() != 'Fuori Produzione']])[".$i."]/a/img/@src");  
$queries[$i] = $xpath->query("(//html/body/div/div/div/section/div/ul/li[div[@class='price' and text() != 'Esaurito' and text() != 'Fuori Produzione']])[".$i."]/a/@href");
```

---

# Descrizione delle fonti - le pagine web

## 4. Preparare la risposta - 5. Fornire la risposta

La risposta sarà un documento XML con la seguente struttura:

**<blocco>**

**<prodotto>**

**<immagine>** l'url all'immagine del prodotto **</immagine>**

**</nome>** il nome del prodotto così come sulla pagina remota **</nome>**

**<prezzo>** il prezzo così come sulla pagina remota **</prezzo>**

**<linkOfferta>** l'url**</linkOfferta>**

**</prodotto>**

...

**</blocco>**

---

# Descrizione delle fonti - le pagine web

## 4. Prepare la risposta - 5. Fornire la risposta

```
127 header("Content-type: text/xml; charset=utf-8");
128 echo "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n";
129 echo "<blocco>";
130 $sadded = 0;
131 for($i = 1; $i <= $count;$i++)
132 {
133     if($sadded <5)
134     {
135         $snome = strtolower($snomi[$i]);
136         if(preg_match($spattern, $snome))
137         {
138             $scode = "<prodotto><immagine>".$simmagini[$i]."</immagine><nome>".trim($snomi[$i])."</nome><prezzo>".trim($sprezzi[$i])."
139             </prezzo><linkofferta>".$squerys[$i]."</linkofferta></prodotto>";
140             echo $scode;
141             $sadded = $sadded +1;
142         }
143     }
144     else
145     {
146         break;
147     }
148 }
149 echo "</blocco>";
```

# Descrizione delle fonti - il DataBase

## Struttura delle tabelle

Nome	Num_click	timestamp	prezzo	url_imm	url_offerta	Hash_link
iPhone 6 128GB Gold MG4E2	10	2015-10-19 13:15:55	789	http://media .glistockisti.i t/media/cata log/product	http://www. glistockisti.it /iphone-6- 128gb-gold- mg4	ebfc86ac35 9b82a757c 23fe7a4709 400

---

# Descrizione delle fonti - il DataBase

## Inserimento database

```
function insertLink($mysqli, $url, $img, $prezzo, $nome) {
    $hash = $mysqli->real_escape_string(md5($url));
    $url = $mysqli->real_escape_string($url);
    $img = $mysqli->real_escape_string($img);
    $prezzo = floatval($prezzo);
    $nome = $mysqli->real_escape_string($nome);

    $sql = "INSERT INTO most_wanted (hashlink, url_offerta, url_img, prezzo, timestamp, num_click, nome_oggetto)
VALUES ('$hash','$url','$img', '$prezzo', NOW(), '1', '$nome') ON DUPLICATE KEY UPDATE timestamp=NOW(), num_click=num_click+1 ;";
```

---

# Descrizione delle fonti - il DataBase

## Ricerca più cliccati

```
function getMostWonted($mysqli) {
    $sql = "SELECT * FROM most_wanted order by num_click desc limit 5";
    if (!$result = $mysqli->query($sql)) {
        // Oh no! The query failed.
        echo "Sorry, the website is experiencing problems.";

        // Again, do not do this on a public site, but we'll show you how
        // to get the error information
        echo "Error: Our query failed to execute and here is why: \n";
        echo "Query: " . $sql . "\n";
        echo "Errno: " . $mysqli->errno . "\n";
        echo "Error: " . $mysqli->error . "\n";
    }

    // $result->num_rows
    return $result;
}
```

# Descrizione delle fonti - API YouTube

Per la ricerca delle recensioni su YouTube è stata necessario:

1. Creare una **developer key**
  2. Scaricare le librerie ***Google\_client*** e ***Google\_YouTubeService***
  3. Configurare i parametri della richiesta quali:
    1. Il prodotto da cercare
    2. Il massimo numero di risultati desiderati
    3. Il region code per la restituzione geolocalizzata dei risultati
      1. Quest'ultimo parametro è successivamente risultato affetto da [bug](#)
-

# Descrizione delle fonti - API YouTube

## Il codice

```
14 $client = new Google_Client();
15 $client->setDeveloperKey($DEVELOPER_KEY);
16
17 $youtube = new Google_YouTubeService($client);
18
19 try {
20     $searchResponse = $youtube->search->listSearch('id,snippet', array(
21         'q' => $_GET['q'],
22         'regionCode' => 'IT',
23         'maxResults' => $_GET['maxResults'],
24     ));
25
26     $videos = '';
27     $channels = '';
28
29     foreach ($searchResponse['items'] as $searchResult) {
30         switch ($searchResult['id']['kind']) {
31             case 'youtube#video':
32                 $videos .= sprintf('<li>%s (%s)</li>', $searchResult['snippet']['title'],
33                     $searchResult['id']['videoId']. "<a href=http://www.youtube.com/watch?v=". $searchResult['id']['videoId']. " target=_blank> Watch This Video</a>");
34                 break;
35             case 'youtube#channel':
36                 $channels .= sprintf('<li>%s (%s)</li>', $searchResult['snippet']['title'],
37                     $searchResult['id']['channelId']);
38                 break;
39         }
40     }
41 }
```

# Schema Globale

- Offerta(nome,link,immagine,prezzo,LinkVideo,fonte)
  - ProdSimili(nome,prezzo,immagine)
  - MostWanted(nome,link,immagine,prezzo,data,numClick,origine)
-

# Query globali

## ○ Ricerca cellulare

- `ricerca_cell(Nome,Prezzo,Link,Imm, LinkVideo,Fonte):-  
Offerta(Nome,Link,Imm,Prezzo,LinkVideo,Fonte).`

## ○ Ricerca prodotti simili

- `ricerca_simili(Nome,Prezzo,Imm):- ProdSimili(Nome,Prezzo,Imm)`

## ○ Ricerca più cliccati

- `most_wanted(Nome,Link,Imm,Prezzo,Click):- MostWanted(Nome,Link,Imm,Prezzo,_,Click,Fonte)`
-

# Gav & Lav

## Mapping Gav

- $\text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{Fonte}) \supseteq$   
 $\text{wireshop}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{'wireshop'}), \text{youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'})$ .
  - $\text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \supseteq$   
 $\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{'stockisti'}), \text{youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'})$ .
  - $\text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \supseteq$   
 $\text{mediaworld}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{'mediaworld'}), \text{youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'})$ .
  - $\text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \supseteq$   
 $\text{pixmania}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{'pixmania'}), \text{youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'})$ .
  - $\text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \supseteq$   
 $\text{mobileshop}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{'mobileshop'}), \text{youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'})$ .
  - $\text{ProdSimili}(\text{Nome}, \text{Prezzo}, \text{Immagine}) \supseteq \text{Stockisti}(\text{N}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{'stockisti'})$
-

# Gav & Lav

## Mapping Lav

- $\text{wireshop}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \subseteq \text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{'wireshop'})$
  - $\text{stockisti}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \subseteq \text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{'stockisti'})$
  - $\text{mediaworld}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \subseteq \text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{'mediaworld'})$
  - $\text{pixmania}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \subseteq \text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{'pixmania'})$
  - $\text{mobileshop}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{Fonte}) \subseteq \text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{'mobileshop'})$
  - $\text{Youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'}) \subseteq \text{Offerta}(\text{Nome}, \text{Link}, \text{Immagine}, \text{Prezzo}, \text{LinkVideo}, \text{Fonte})$
-

# Gav & Lav

## reformulazione query ricerca cellulare

- $\text{ricerca\_cell}(\text{Nome}, \text{Prezzo}, \text{Link}, \text{Imm}, \text{LinkVideo}, \text{Fonte})$ :-

$( \text{wireshop}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{Fonte}) \vee$

$\text{stockisti}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{Fonte}) \vee$

$\text{mediaworld}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{Fonte}) \vee$

$\text{pixmania}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{Fonte}) \vee$

$\text{mobileshop}(\text{Nome}, \text{Link}, \text{Imm}, \text{Prezzo}, \text{Fonte})) \wedge$

$\text{youtube}(\text{Nome}, \text{LinkVideo}, \text{'IT'}, \text{'5'})$ .

---

# Gav & Lav

reformulazione query più cliccati

- `most_wanted(Nome,Link,Prezzo,Click):- db(Nome,_,Click,_),  
ricerca_cell(Nome,Prezzo,Link,_,_), Click>50.`

## Più cliccati

- SAMSUNG SM-N910F Galaxy Note 4 White - 499.99€
  - Apple iPhone 6 Plus 128GB Oro - 793€
  - Nokia Lumia 510 Giallo - 129€
  - Huawei Honor 6 Plus 32GB Dual SIM PE-TL10 Bianco - 383€
  - LG G4 H815 Bianco Oro - 445€
-

# Gav & Lav

## reformulazione query prodotti simili

- `ricerca_simili(Nome,Prezzo,Imm):- ricerca_cell(Nome,Prezzo,_,Imm,_,'stockisti')`,

### Prodotti simili



iPhone 6 128GB Gold a 689,00€



iPhone 6S 64GB Gold a 719,00€



iPhone 6S Plus 16GB Gold a 759,00€

# Le Tecnologie

- **jQuery + MixItUp:** librerie Javascript utilizzate nel font-end del sito, per mostrare i risultati delle ricerche



## MixItUp

- **XML e JSON:** utilizzate per trasferire i dati tra i vari layer del programma e per immagazzinare i dati non strutturati

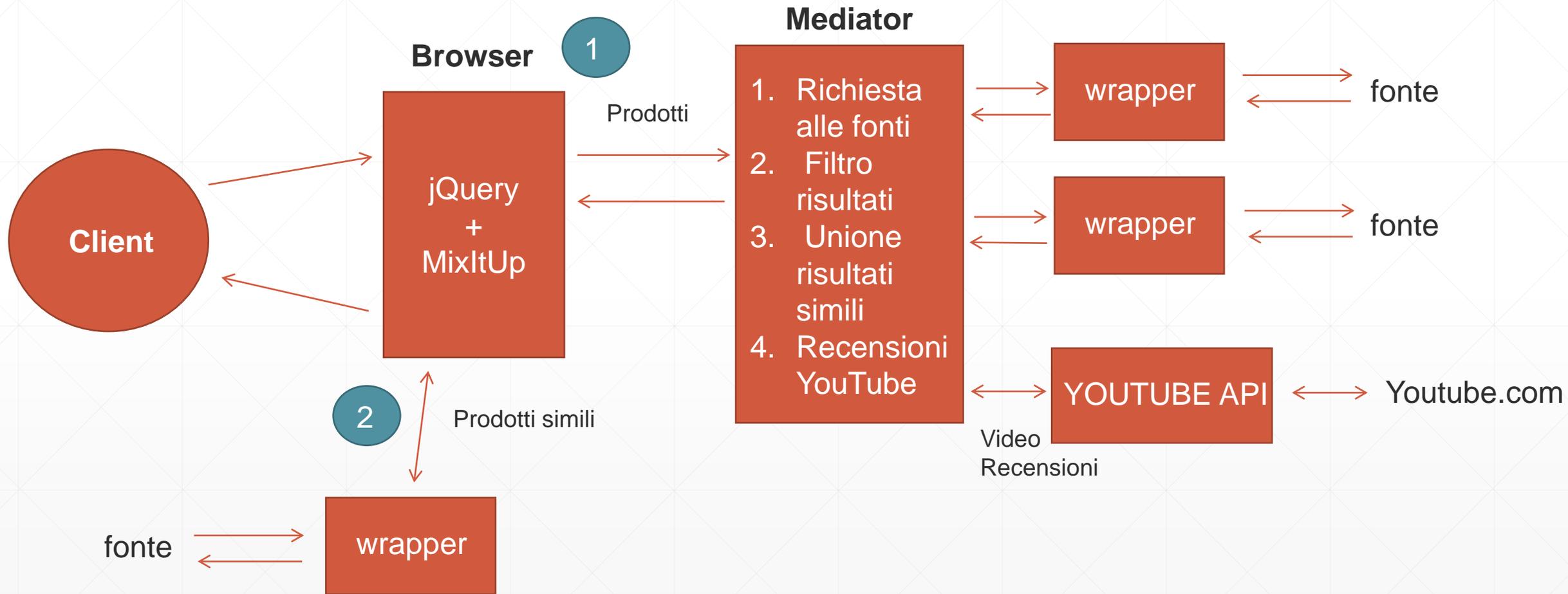


- **PHP:** utilizzato per la realizzazione dei wrapper e per l'incrocio dei dati



# Le Tecnologie

## Iterazioni Client-Server



# Le Tecnologie

## Iterazioni Client-Server

- Mediator PHP:
    - I dati dai wrapper arrivano sotto forma di XML
    - Essi vengono poi filtrati e puliti
    - Una volta puliti vengono combinati in base alla similitudine del nome e viene generato un JSON
    - Nel JSON vengono inserite le video-recensioni da Youtube, prese tramite API
    - Il JSON viene salvato per le future query e viene mandato al client per mostrare i risultati
  - Pagina Web, usata come integratore di dati:
    - Utilizza Bootstrap per l'estetica e la grafica responsive
    - Utilizza MixItUp in modo per poter filtrare e/o ordinare i dati a seconda delle fonti e del prezzo
    - Quando vengono mostrati i risultati, vengono mostrati anche i prodotti simili
-

# Conclusioni

- Ogni fonte è stata volutamente trattata a parte
    - Totale indipendenza delle fonti
    - Maggiore facilità di manutenzione
    - Maggiore facilità di nuove implementazioni
  - Al crescere del progetto verranno aggiunte nuove fonti
    - La concorrenzialità va a favore di PriceFinder
  - Nuove features previste:
    - Storico dei prezzi
    - Storico release dei firmware ( iOS 7/8/9 ... Android KitKat/JellyBean/Marshmallow ... )
-

**Fine!**

